Exhibit D

Scalable Intrusion Detection for the Emerging Network Infrastructure

Objective

What: This three-year DARPA-funded project is to design and develop a software system for protecting against intruders from breaking into network routers, switches, and network management channels. The project is a joint collaboration between MCNC and North Carolina State University (NCSU).

Why: Given the increasing popularity of the Internet, intrusion incidents are becoming common events of life. Attacks on the network infrastructure has the potential of disrupting a large scale of information services on which the national defense and economy may depend. Despite the best efforts of the protocol designers, implementors, and system administrators, it is prudent to assume that attacks will occur and some, unfortunately, will succeed. Therefore, it is vitally important to develop means to automatically detect and respond to these attacks in order to maintain these critical information services.

Approach

In this project, we will design, implement, and integrate intrusion detection techniques based on statistical and logical analysis of network routing and management protocols to construct a scalable distributed intrusion detection system for the emerging internetwork environment. At the top level, the system consists of local detection subsystems and remote management application subsystems. The integration of these two subsystems will be mapped onto the SNMP standard management framework.

A local subsystem has three major components: rule-based prevention module, protocol-based detection module, and statistical analysis detection module. As a gate-keeper, the prevention module intercepts and filters all incoming packets according to a small set of rules. It conducts a quick check to see whether an incoming packet violate general security guidelines or special administrative security concerns. A second component of the system uses logical analysis of protocol operation. This technique detects intrusion by monitoring the execution of protocols in a router/switch and triggering an intrusion alarms when an anomalous state is entered. The statistical-based approach is founded on the contention that network routing and management protocols exhibit certain behavioral signatures. Any behavior deviating from the normal signature will be considered as an anomaly and appropriate alarms can be triggered.

The detection functions of a local subsystem are complementary in nature in terms of their capabilities and their response times. The rule and protocol based approach is meant to analyze and detect known vulnerabilities. On the other hand, the statistical analysis is intended to uncover those attacks that cannot be prevented by a set of rules embedded in a rule-based component or cannot be detected by security analysis conducted through protocol-based approach. As far as response time is concerned, the statistical approach requires an observation window to determine whether the target is anomalous. The protocol-based and, especially, the rule-based mechanisms will be able to detect the targeted intrusions with relatively low latency.

For demonstration purposes, we will implement simple network management applications for accessing and coordinating local detection information. The choice of using SNMP as the information exchange protocol was based on the fact that it is standardized and any other security applications based on SNMP may potentially interoperate with our system with relative ease.

To evaluate the system design and implementation, we will develop a set of attacks and use them to exercise our system by attacking nodes within a testbed network. These tests will allow us to measure the run-time overhead introduced by the intrusion detection system. After the validation process, we expect to deploy and evaluate the system in an operational network.

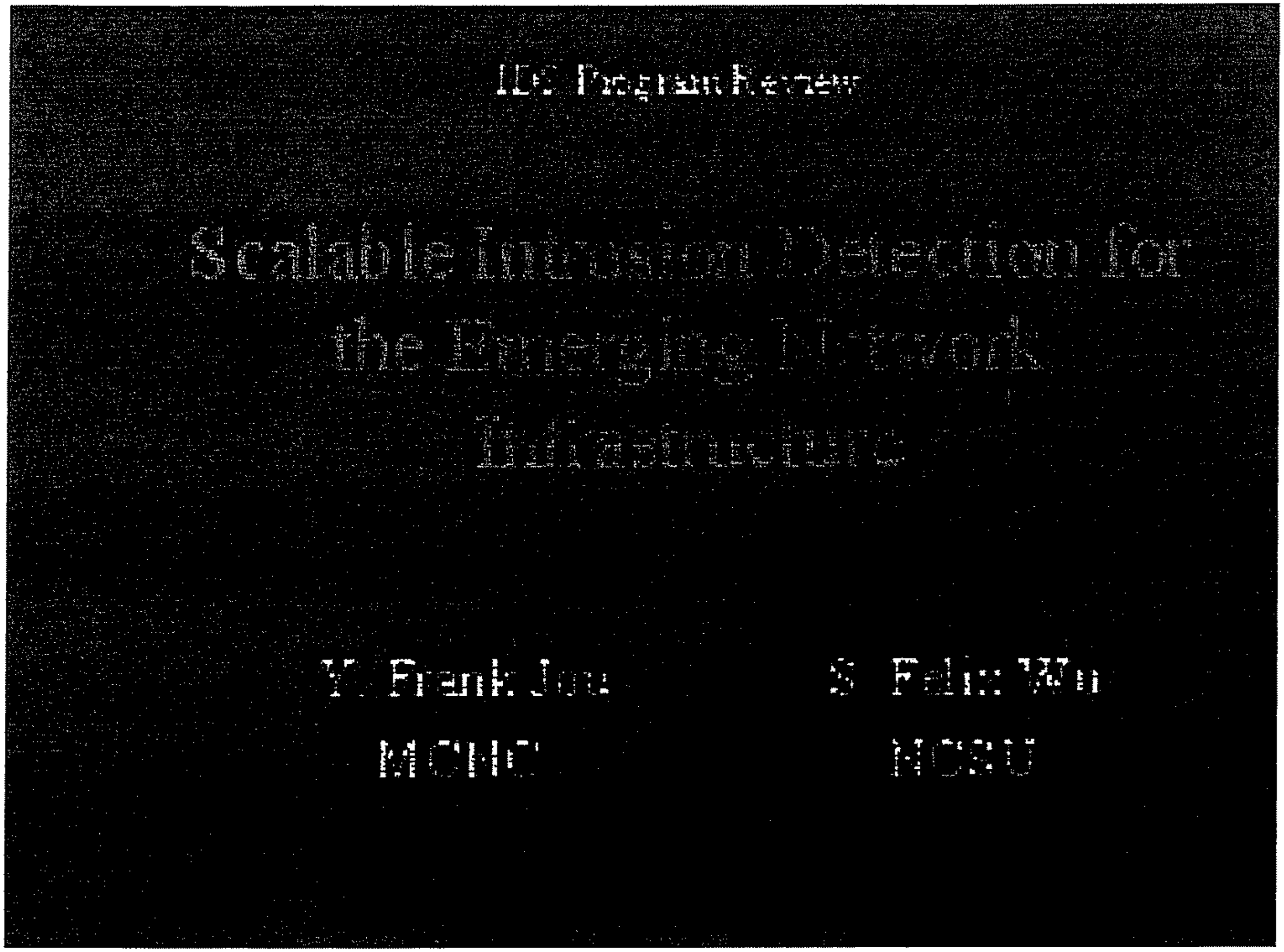
Related Information

- Project Viewgraph (powerpoint)
- Architecture Design Report (postscript)
- Project Update Viewgraph (at SRI, July 97, powerpoint)
- Project Update Viewgraph (at Annapolis, MD, Feb. 98, powerpoint)

- Contact: Y. Frank Jou, Email: jou@mcnc.org

MCNC
Post Office Box 12889
Research Triangle Park
North Carolina 27709-2889

Last Modified: September 24, 1997



IDS Program Review

Scalable Intrusion Detection for the Emerging Network Infrastructure

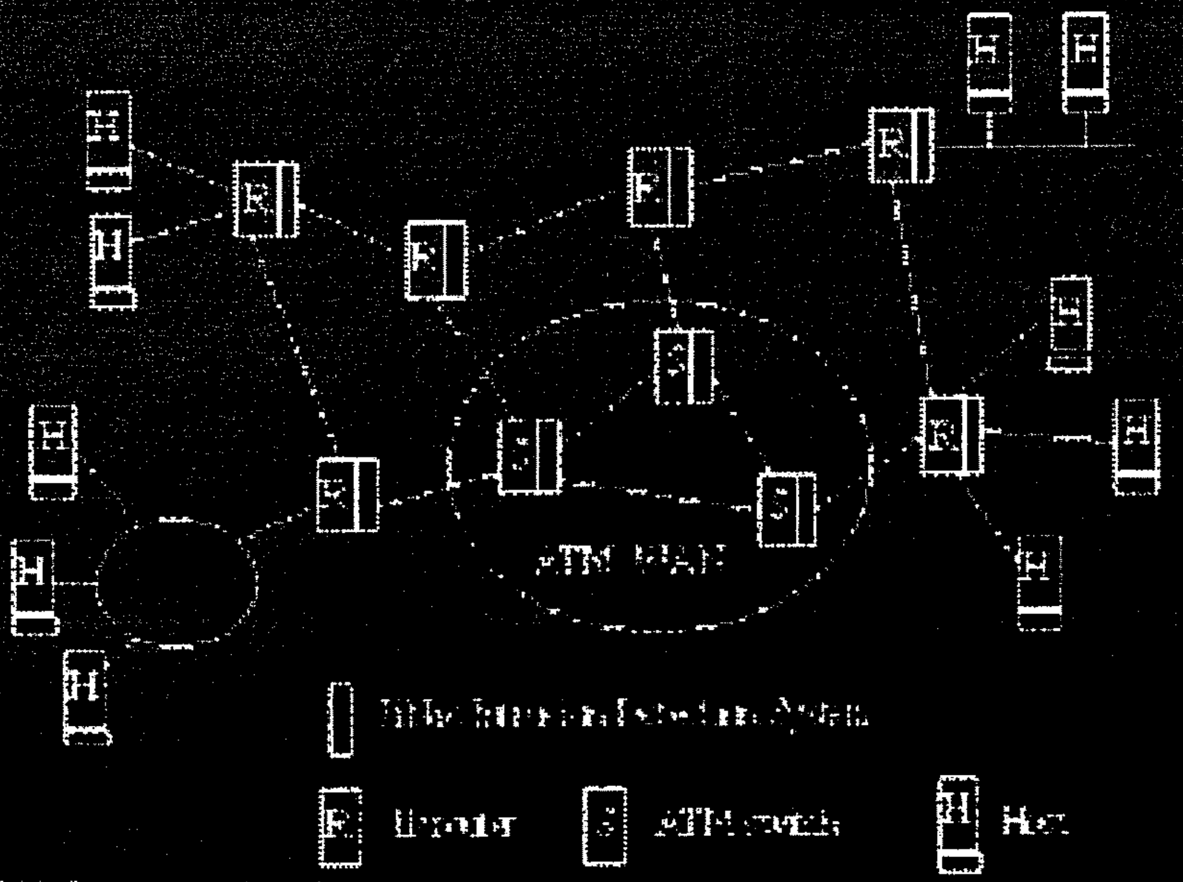
Y. Frank Jou
MCNC

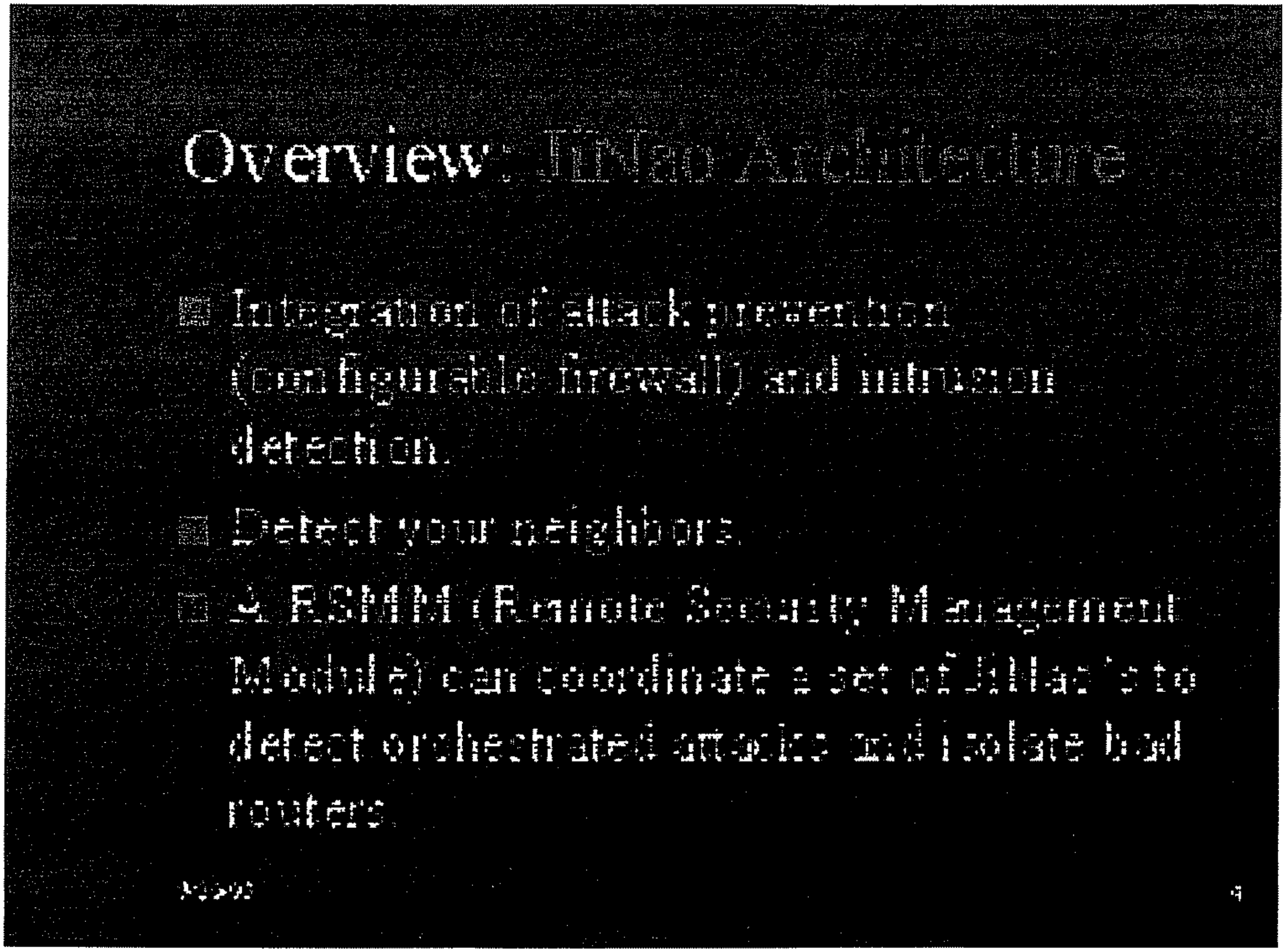
S. Felix Wu
NCSU

Project Update

- Overview
- System Architecture Design
- Routing Testbed Configuration
- Routing Traffic Statistical Profiles
- Routing Attacks
- What's next

Overview: Target Environment



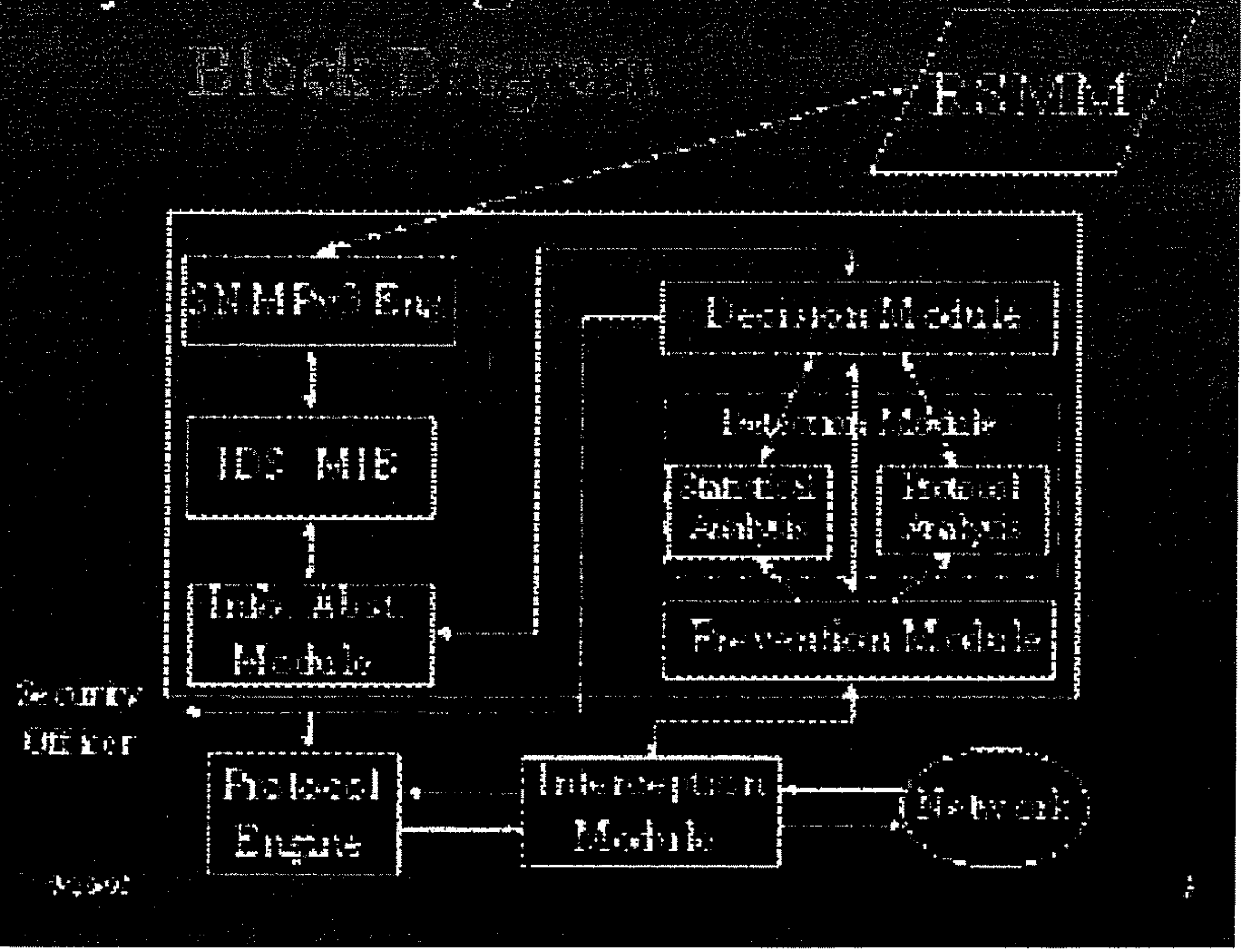


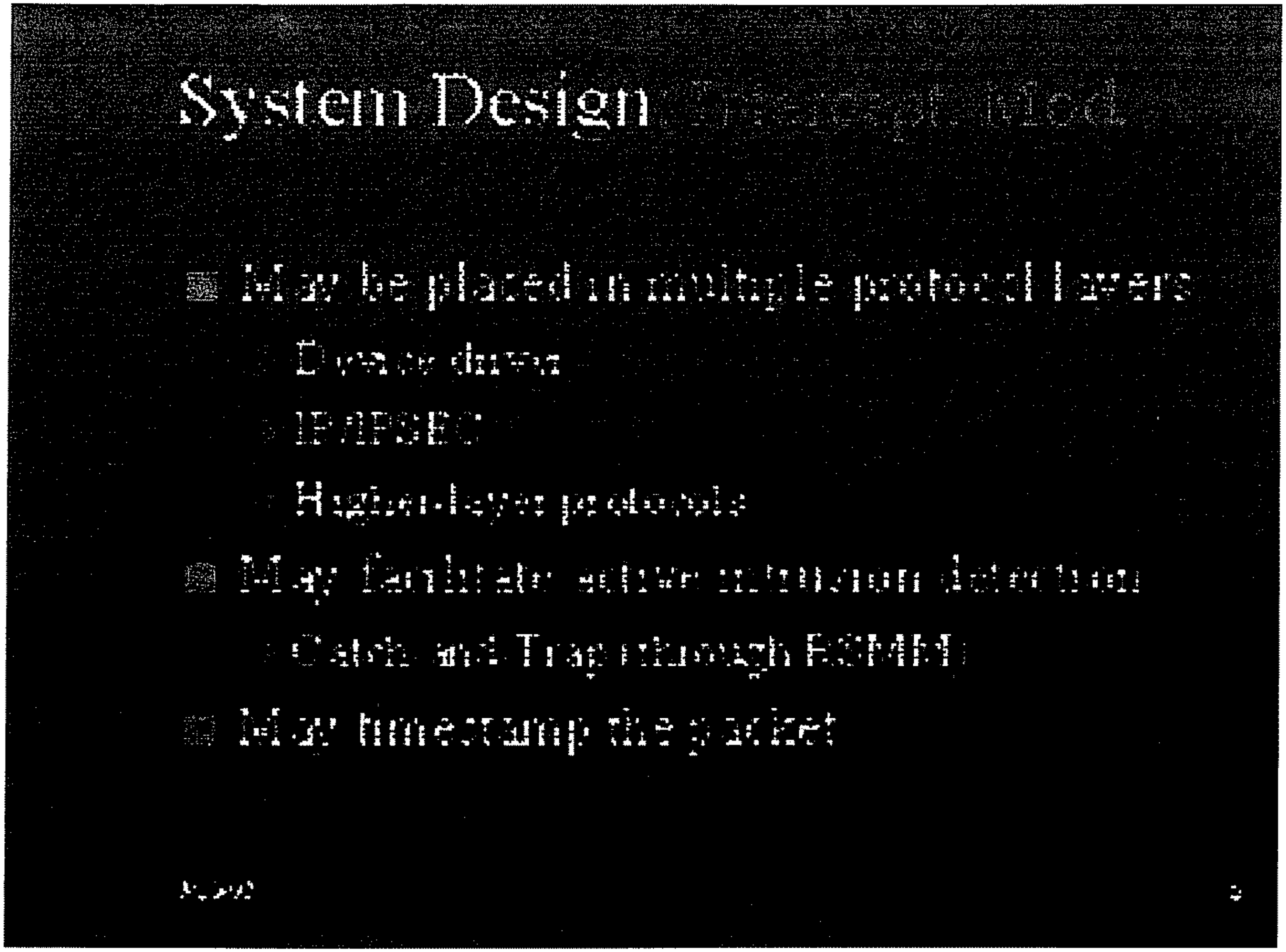
Overview: JiNao Architecture

- Integration of attack prevention (configurable firewall) and intrusion detection.
- Detect your neighbors.
- A RSMM (Remote Security Management Module) can coordinate a set of JiNao's to detect orchestrated attacks and isolate bad routers.

System Design

Block Diagram





System Design Encrypt Mod.

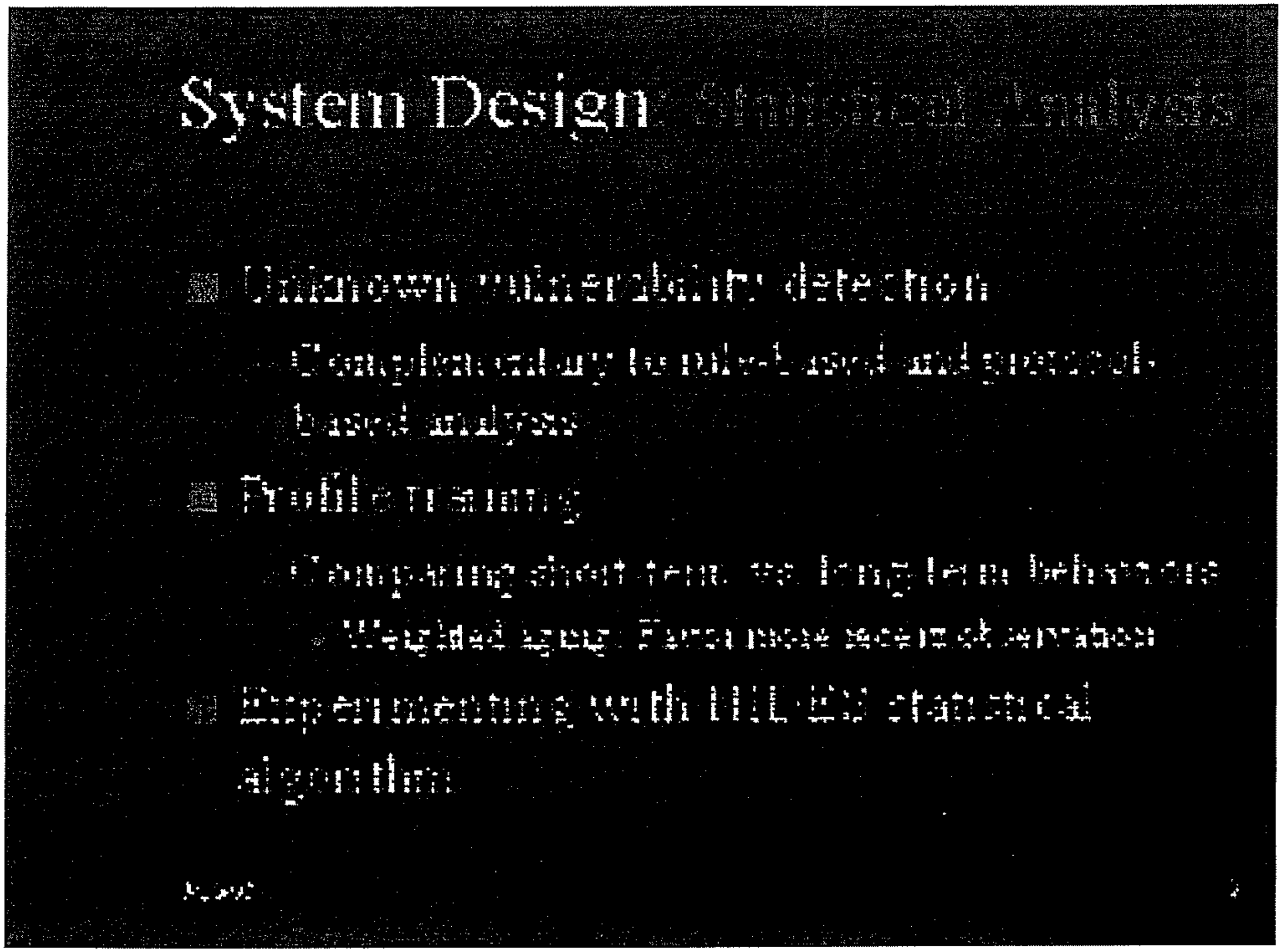
- May be placed in multiple protocol layers
  - Device driver
  - IP/IPSEC
  - Higher-layer protocols
- May facilitate active intrusion detection
  - Catch and Trap through ESMM
- May timestamp the packet

System Design: Prevention Mod[illegible]

- Prevention Layer: Go or No-Go
  - Quick response
- Extraction Layer
  - FDU format conversion
  - Multiple inter-operating points correlation
    - e.g. The SNMP trap and FDU is from [illegible] and [illegible]

System Design: Protocol Analysis

- Maintain a set of Finite State Machines
  - One FSM for each identified intrusion
- Provide extensibility
  - Reconfigurable at Run-Time
    - Table-driven implementation of FSMs
    - With a generic driver routine
- Use Concurrency Workbench (CWB) to produce the FSMs



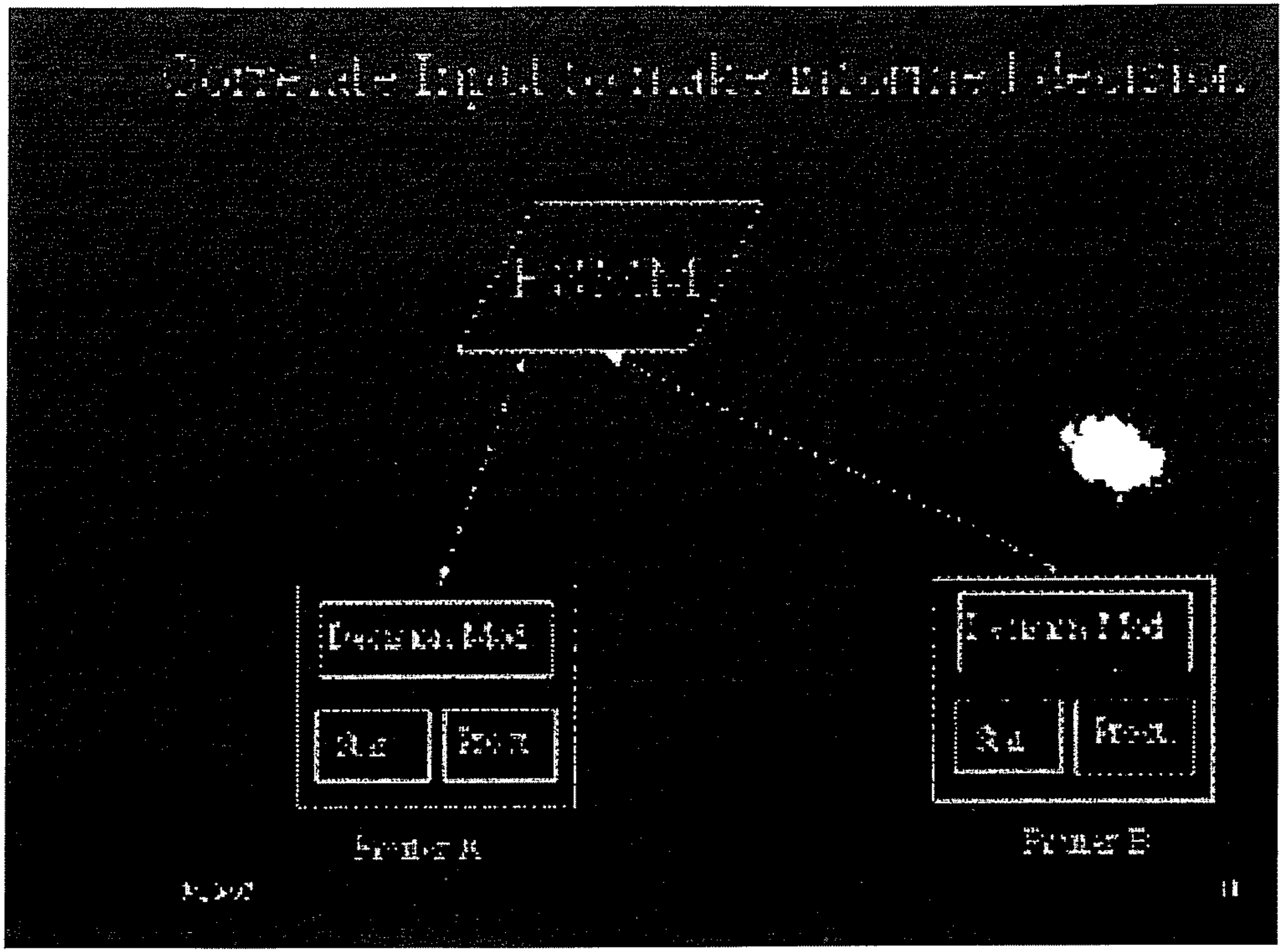
System Design: Statistical Analysis

- Unknown vulnerability detection
  - Complementary to rule-based and protocol-based analyses
- Profile training
  - Comparing short term vs. long term behaviors
    - Weighted aging: Favors more recent observations
- Experimenting with NIDES statistical algorithm

SRI

System Design: Decision Mod.

- Make decisions on intrusion based on input from Prev/Detec Modules and RSMM
- Provide information for the IAM (RSMM)
- Propagate global information to the Prev/Detec Modules
- Notify security officer of faults/intrusion



Correlate Input to make informed decision.

System Design: Info. Abst. Mod.

- Detection Info. aggregation/MIB-ification
  - Run-length coding for data reduction for repeated normal report or persistent fault
- Periodic checking and propagation of global information
- Scope of Impact representation
  - Topological info. on all the affected routers through graph representation (GrIDS?)

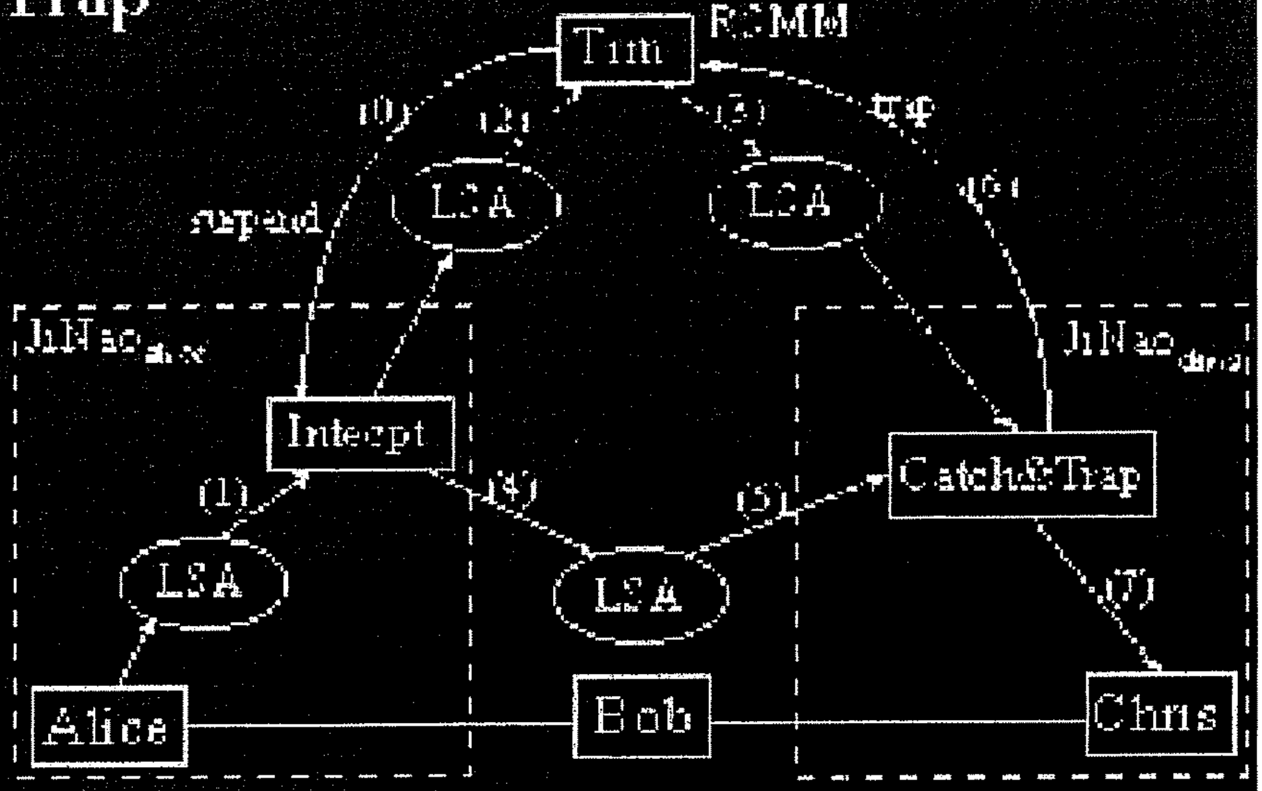
System Design

- Rule/FSM configuration and statistical parameter specification
- Local detection results
- Detection notifications
- Security control
- Log access

System Design: [illegible]

- SNMPv3 based management applications
- Access JiNao MIBs and correlate detection results
- Example: active intrusion detection (Catch and Trap)

Active Intrusion Detection: Catch & Trap



15

System Design: Interfaces

- Information exchange is done via message passing
- Authentication is provided if necessary
- Separate input queue for facilitating priority mechanisms

Testbed Configuration

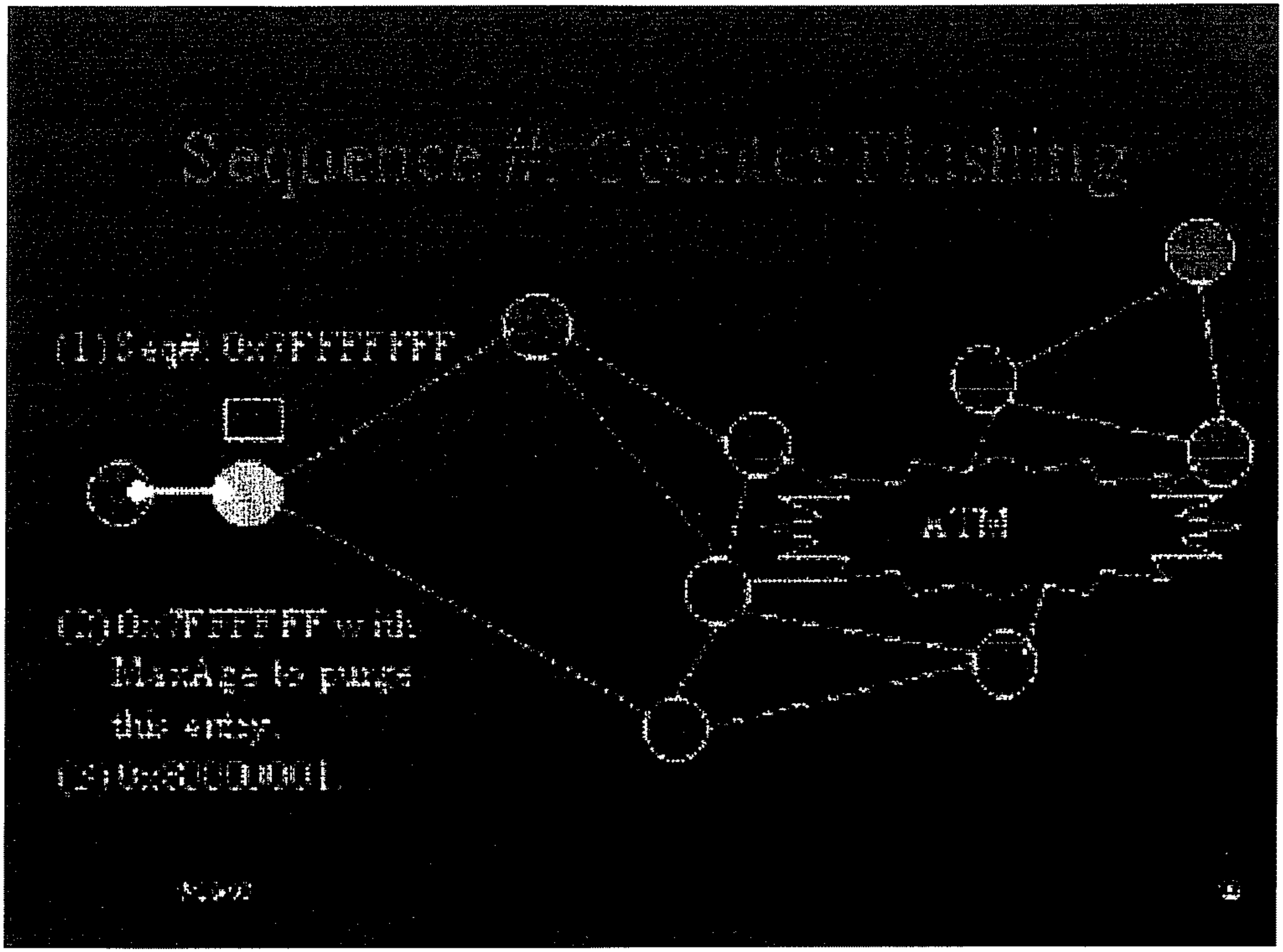
- Two routing testbeds (autonomous systems. AS): MCNC & NCSU
- Each has three to four areas
- Allow independent code development
- Will be linked together to experiment ASBR attacks (only AS-external LSAs are flooded throughout the entire AS)
- (Ref. configuration in file "testbed.ps")

Routing Statistical Profiles

- Hello packets: stable (like step-function)
- Database Description and LS Request packets: rare events (only for forming adjacencies)
- LS Update and LS Ack: periodic in about every 30 min (LSRefreshTime: 30 min, MinLSInterval: 5 sec)
- (Look for four other postscript files, two were normal, two were under attack)

Routing Attacks

- Hit-and-Run attack: hard to detect/isolate
  - Inject one or very few bad packet causing long term damage.
- Persistent Attack
  - The intruder has to continuously inject attack packets
- Attack Experiments
  - Max Sequence Number attack: simple, bugs
  - MaxAgeDiff attack: weak checksum algo [illegible]

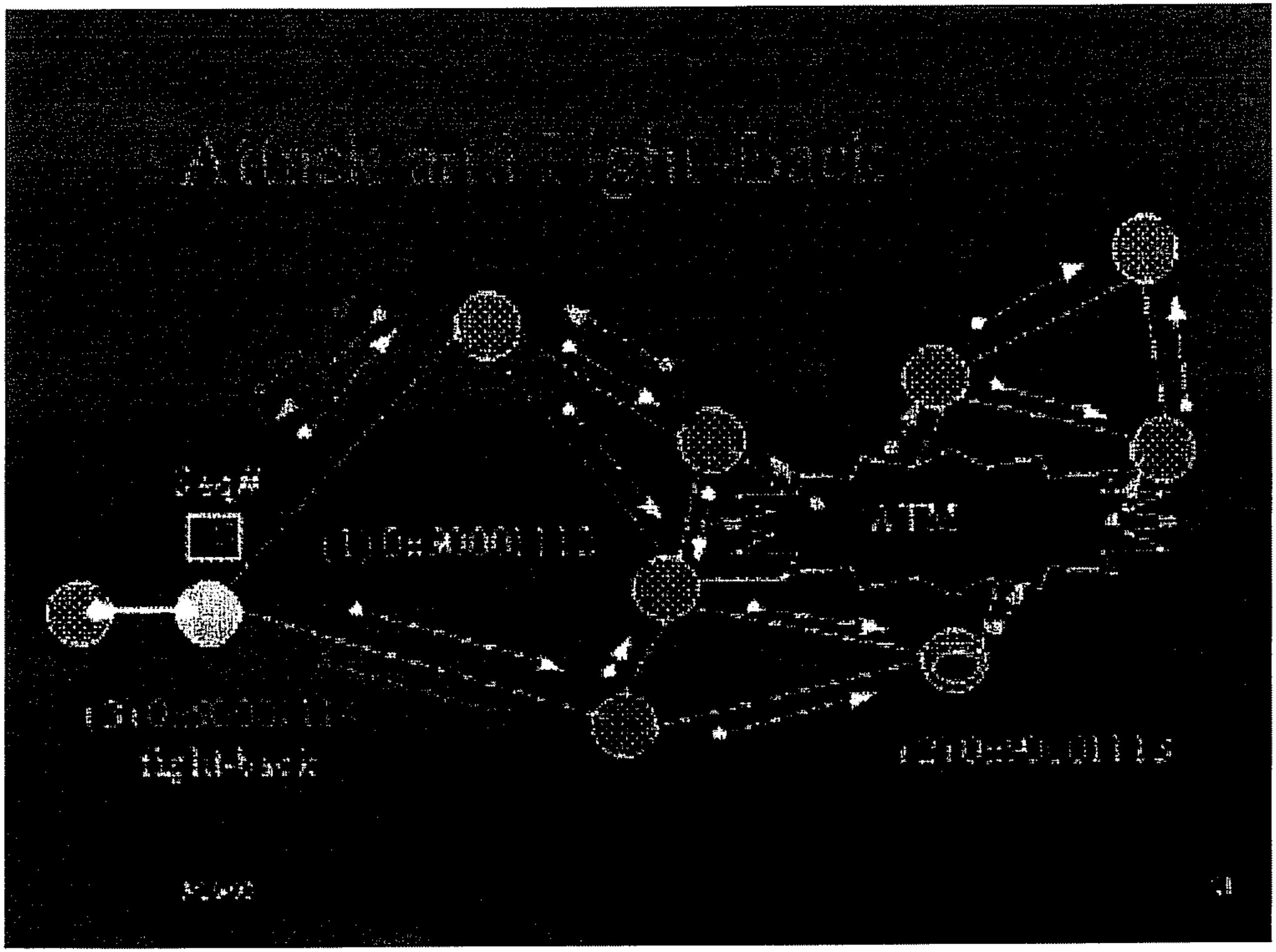


Sequence # Counter Flushing

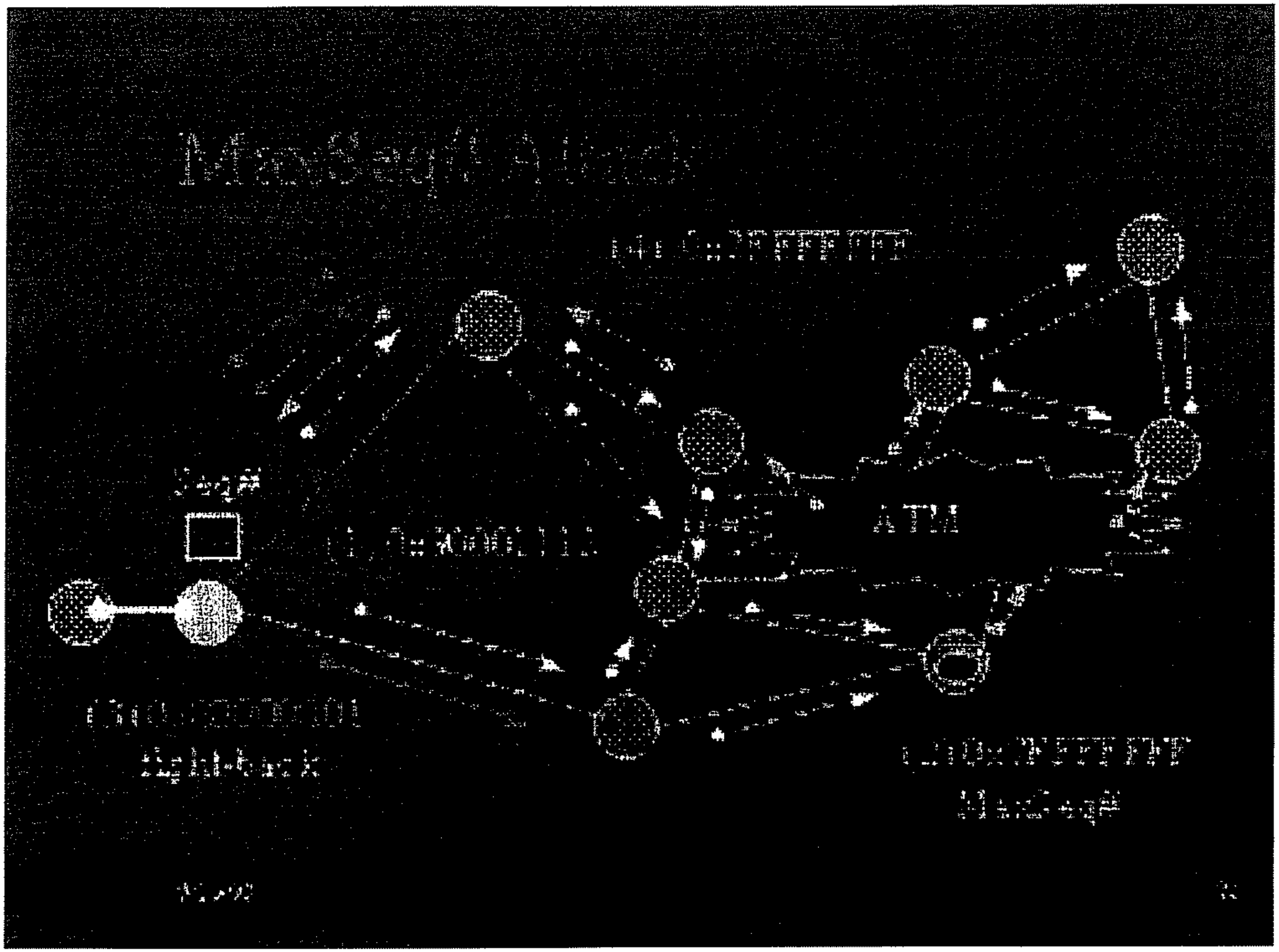
(1) Seq#: 0x7FFFFFFF

(2) 0x7FFFFFFF with
MaxAge to purge
this entry.

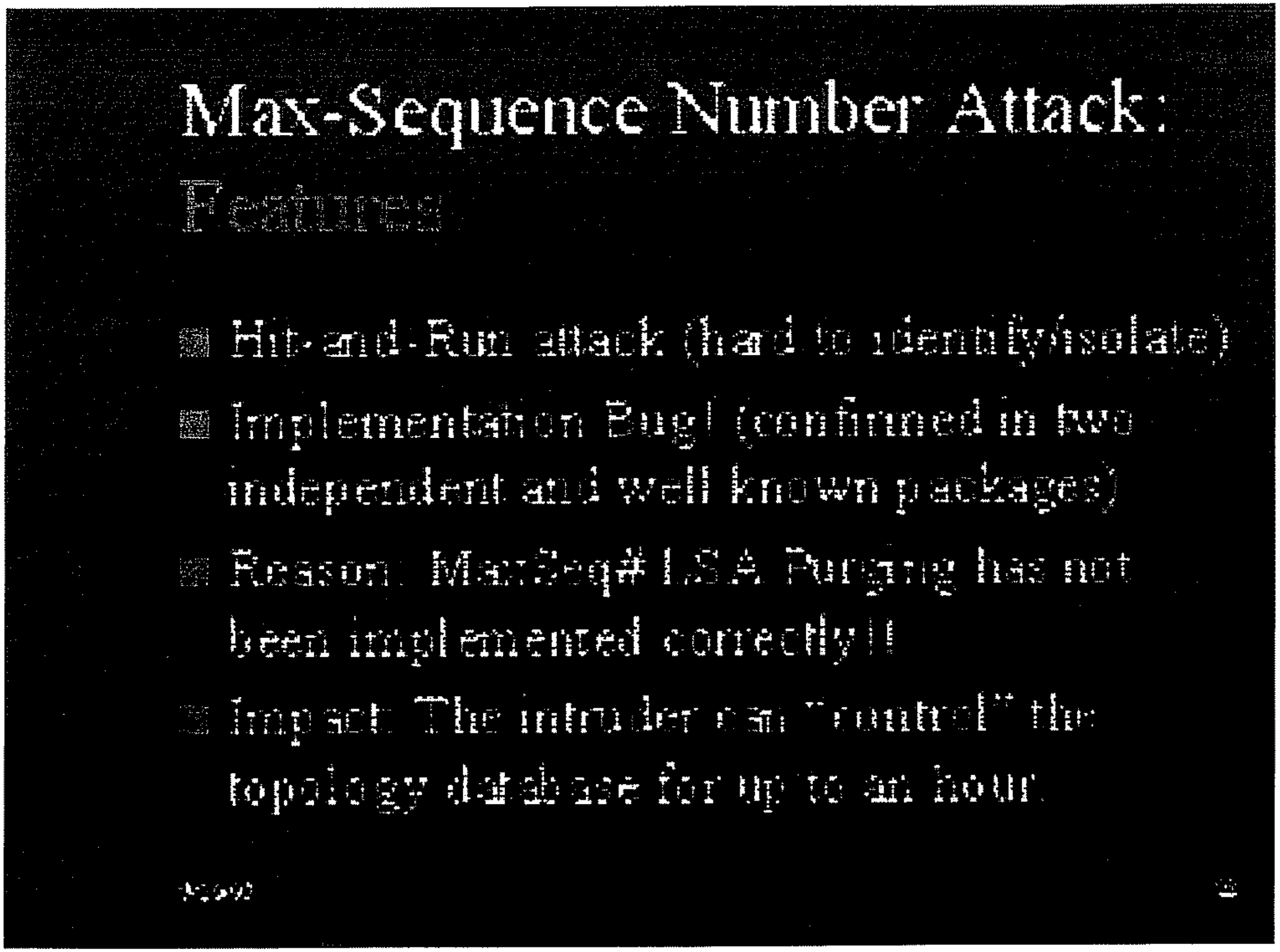
(3) 0x80000001



Attack and Fight-Back



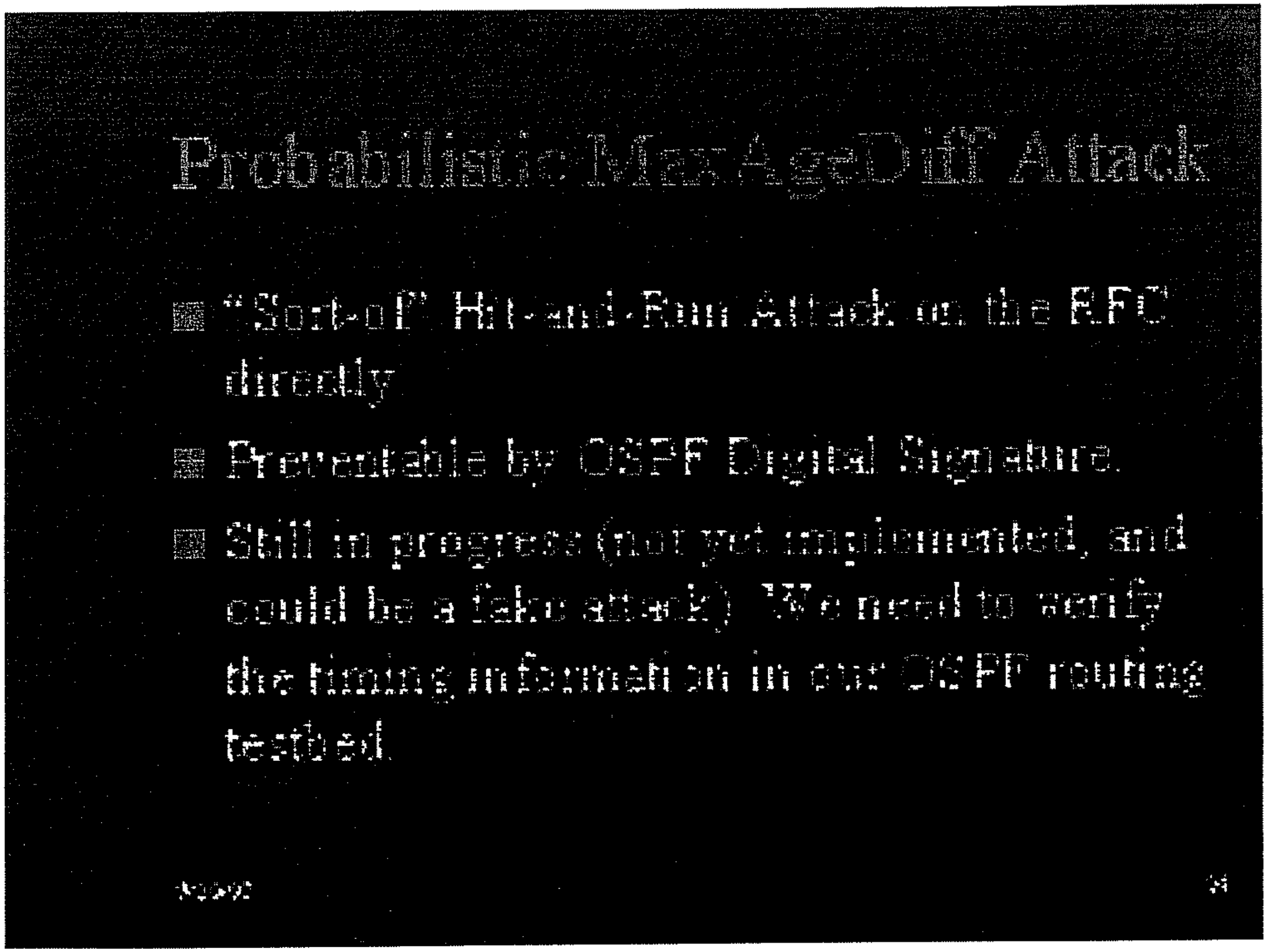
MaxSeq# Attack



Max-Sequence Number Attack:

Features

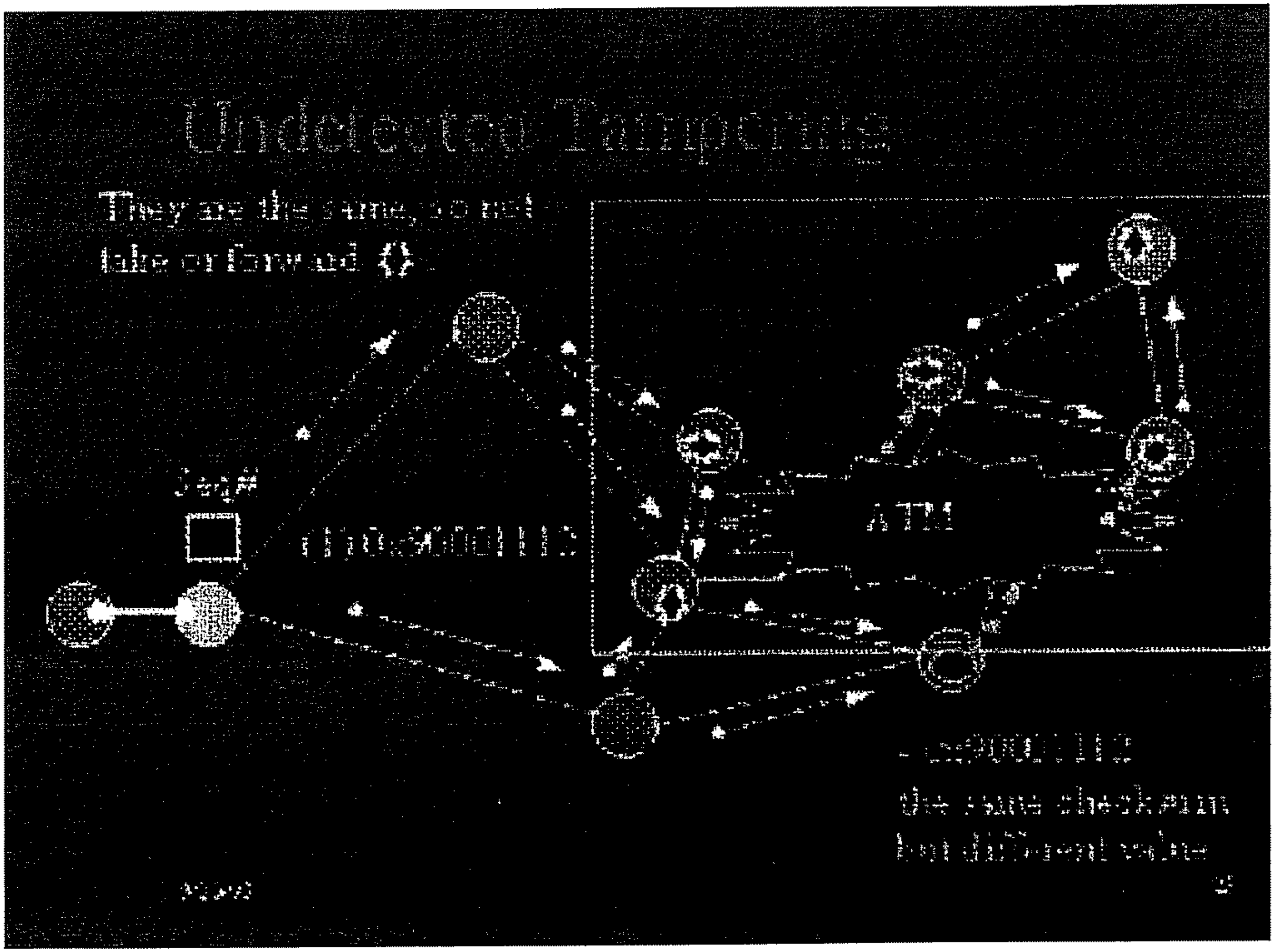
- Hit-and-Run attack (hard to identify/isolate)
- Implementation Bug! (confirmed in two independent and well known packages)
- Reason: MaxSeq# LSA Purging has not been implemented correctly!!
- Impact: The intruder can "control" the topology database for up to an hour.



Probabilistic MaxAgeDiff Attack

- "Sort-of" Hit-and-Run Attack on the RFC directly
- Preventable by OSPF Digital Signature.
- Still in progress (not yet implemented, and could be a fake attack). We need to verify the timing information in our OSPF routing testbed.

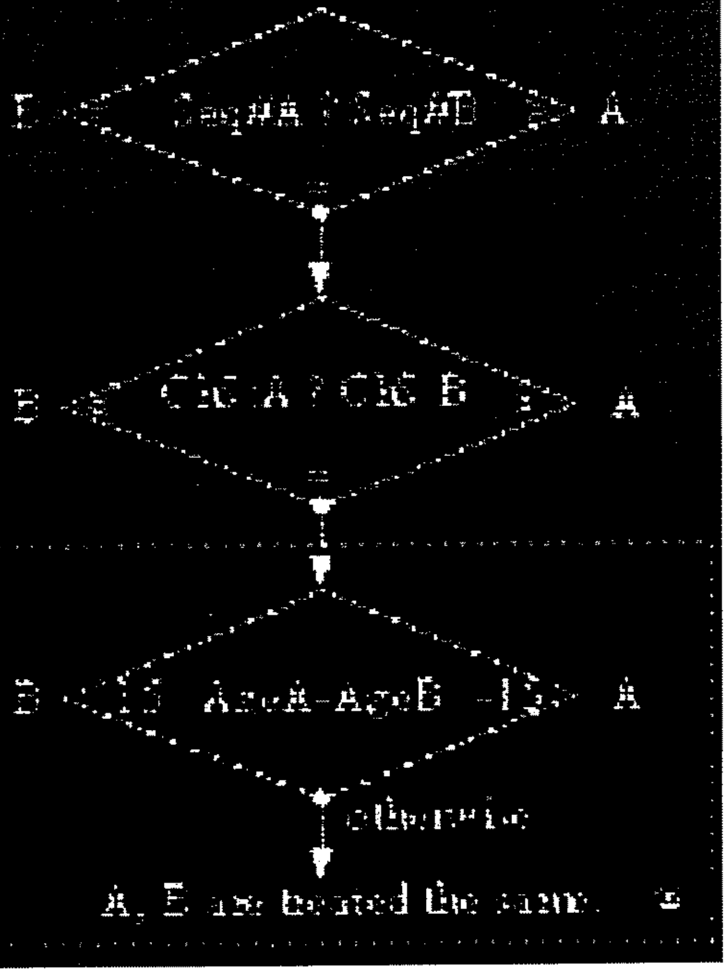
Undetected Tampering



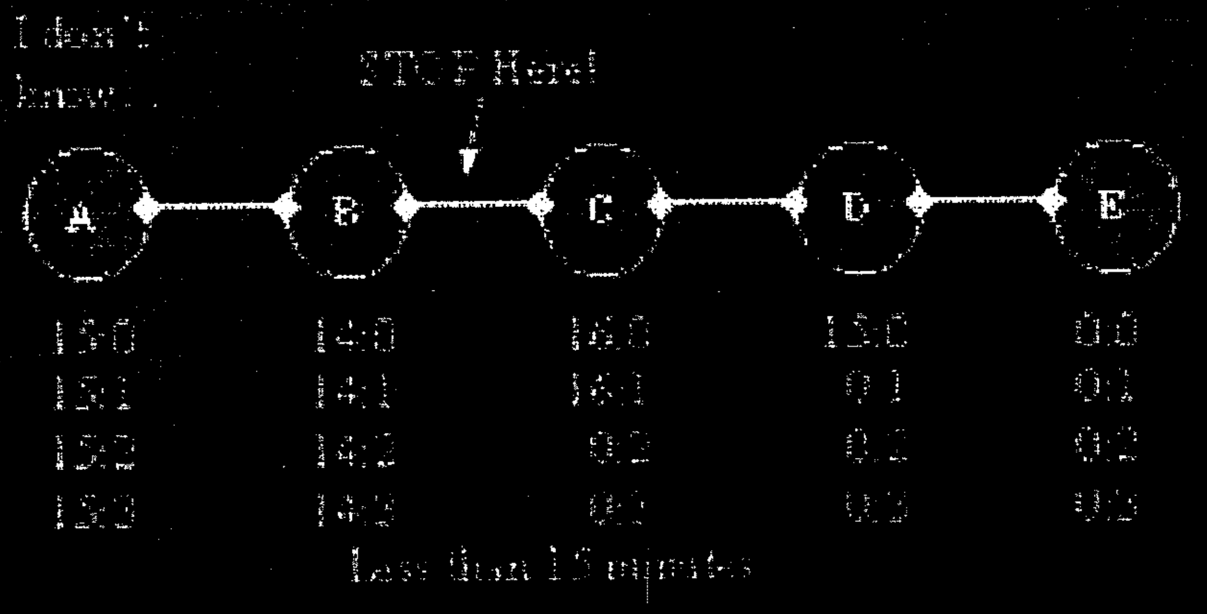
They are the same, so not take or forward {}

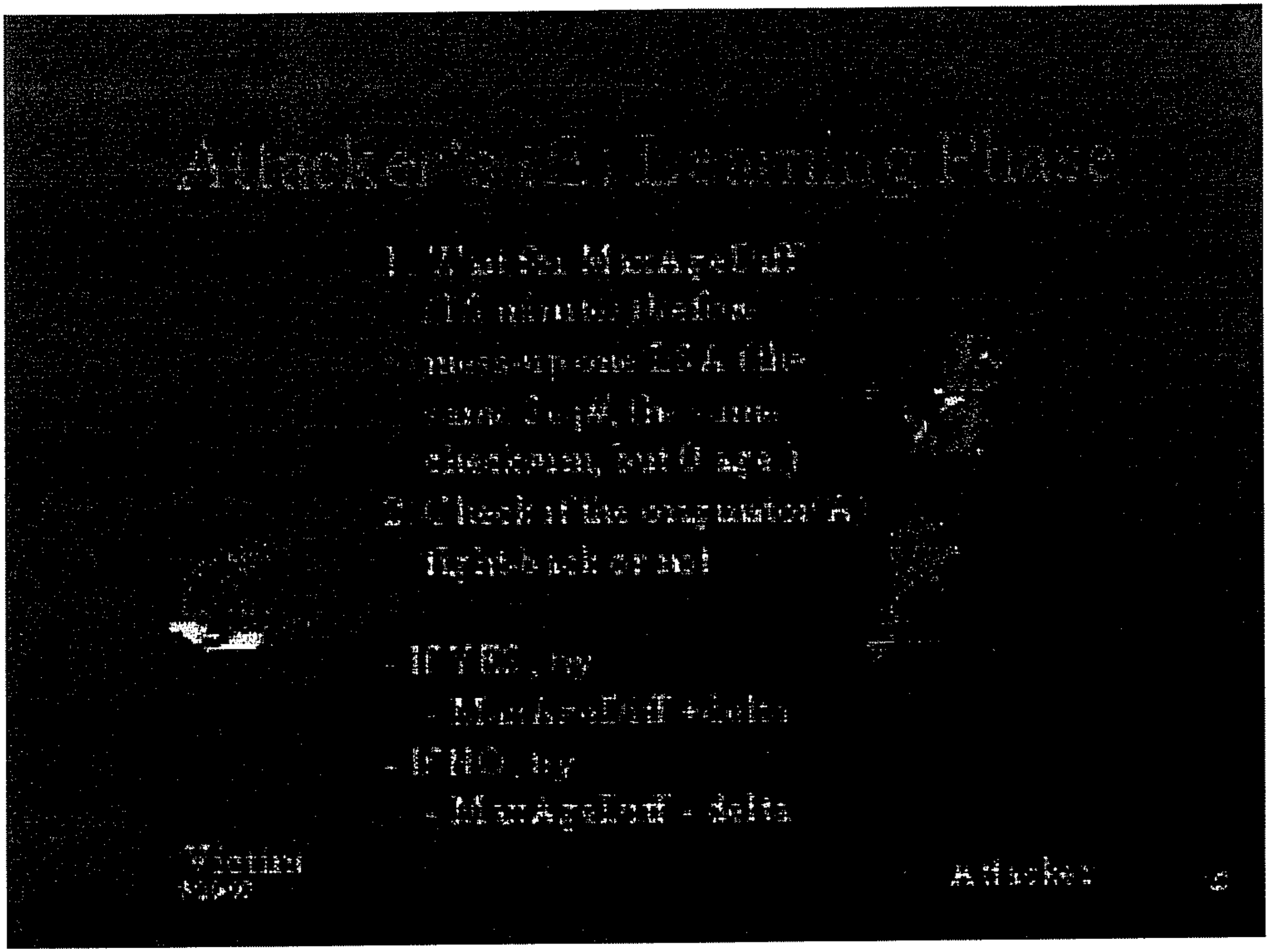
– 0x9000112
the same checksum
but different value

Fresher LSA?


B
<
Seq#A ? Seq#B
>
A
=
B
<
Chk A ? Chk B
>
A
=
B
< -15
AgeA-AgeB
15 >
A
otherwise
A, B are treated the same.

Linear Case: LSA Age

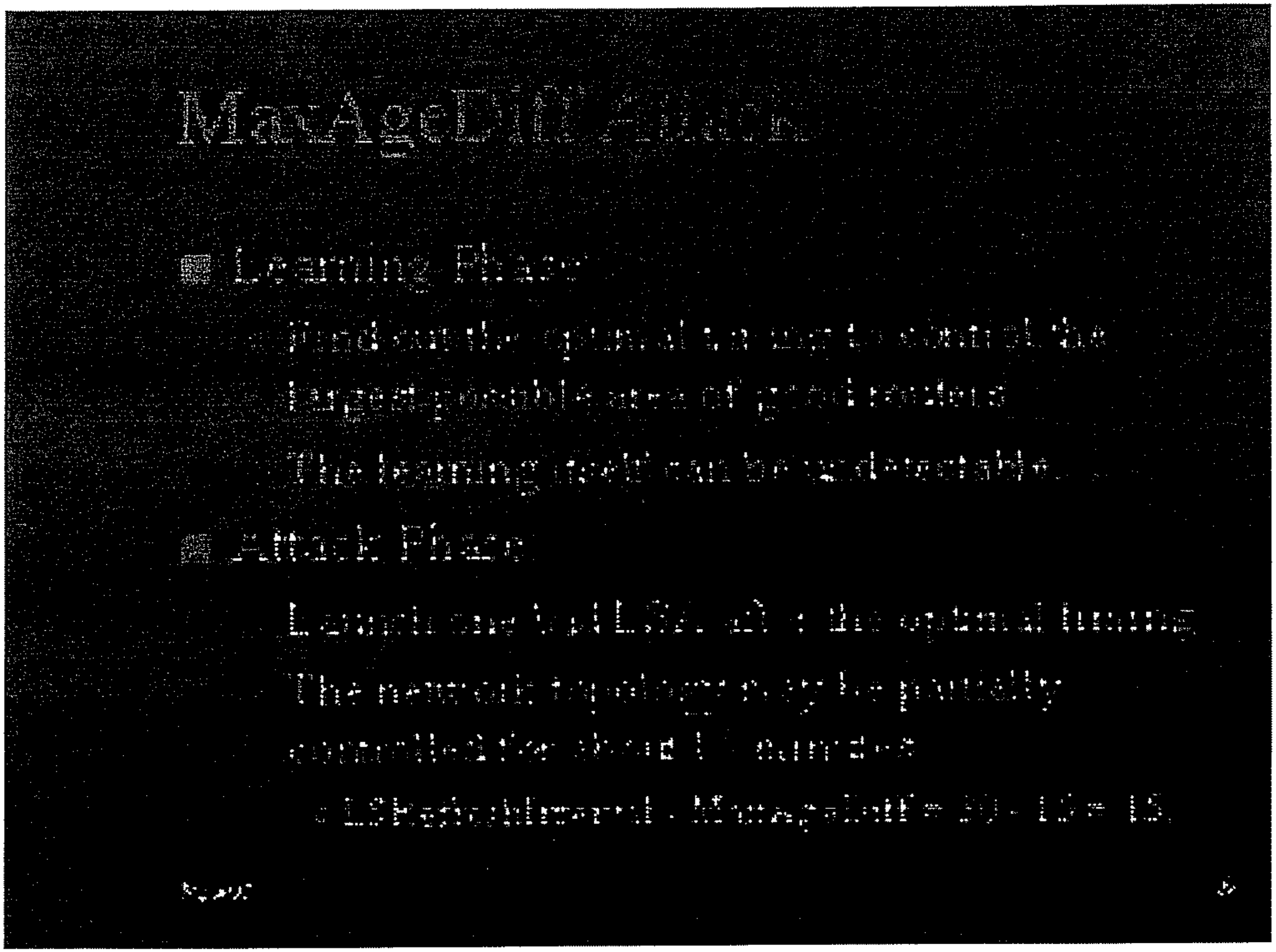




Attacker's (2): Learning Phase

1. Wait for MaxAgeDiff (15 minutes) before mess-up one LSA (the same Seq#, the same checksum, but 0 age)
2. Check if the originator R fight-back or not

- If YES, try
  - MaxAgeDiff + delta
- If NO, try
  - MaxAgeDiff - delta



MaxAgeDiff Attack

- Learning Phase
  - Find out the optimal timing to control the largest possible area of good routers
  - The learning itself can be undetectable.
- Attack Phase
  - Launch one bad LSA at the optimal timing
  - The network topology may be partially controlled for about 15 minutes
    - LSRefreshInterval - MaxAgeDiff = 30 - 15 = 15.

What's Next:

- Implement the module code
- Collect more routing traffic statistical profiles
- Conduct code analysis on GateD (OSPF portion)
- Construct simulator to understand the impact of attack on a large scale network